



No. 2003–105

**SEQUENCING GAMES WITH CONTROLLABLE  
PROCESSING TIMES**

By S. van Velzen

November 2003

ISSN 0924-7815

# Sequencing games with controllable processing times

BAS VAN VELZEN

CentER and Department of Econometrics and Operations Research,  
Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands.  
E-mail: S.vanVelzen@uvt.nl

## Abstract

In this paper we study a class of cooperative sequencing games that arise from sequencing situations in which the processing times are not fixed. We show that these games are balanced by obtaining two core elements that depend only on the optimal schedule for the grand coalition. Furthermore we show that, although these games are not convex in general, many marginal vectors are core elements. We also consider convexity for special instances of the sequencing situation.

*Keywords:* Cooperative games, sequencing situations, controllable processing times

## 1 Introduction

In one-machine sequencing situations each agent has one job that has to be processed on a single machine. Each job is specified by a processing time, and costs are assumed to be linear in the completion time. Furthermore, an initial order on the jobs is given. The objective is to find a processing order that minimizes total cost. Once this optimal order is obtained, the question arises how to allocate the total cost savings to the agents. Curiel et al. [4] consider this problem by introducing cooperative sequencing games. They show that these games are convex, and hence possess core allocations. Furthermore they characterize an allocation rule that divides the cost savings obtained by complete cooperation.

Nowadays, a variety of sequencing games exist in cooperative game theory. Many of these sequencing games arise from extended classes of one-machine sequencing situations. Hamers et al. [7] study balancedness of sequencing games that arise from situations in which ready times are imposed on the jobs. Similarly, Borm et al. [2] investigate balancedness in case due dates are included and Hamers et al. [9] show that if chain precedences are imposed on the jobs, and the initial order is a concatenation of these chains, then the corresponding sequencing game is convex.

A different approach in extending the class of sequencing situations of Curiel et al. [4] is taken by Van den Nouweland et al. [14], Hamers et al. [8] and Calleja et al. [3]. They investigate balancedness of sequencing games arising from situations with multiple machines.

In this paper we treat sequencing situations in which the processing times are not fixed, so called sequencing situations with controllable processing times. In reality processing jobs does not only require a machine, but also additional resources such as manpower, funds, etc. This implies that jobs can be processed in shorter or longer durations by increasing or decreasing these additional resources. Of course, deploying these additional resources entails extra costs, but these extra costs might be compensated by the gains from job completion at an earlier time. Sequencing situations with controllable processing times are investigated in, among others, Vickson [16; 17], and Alidaee and Ahmadian [1]. An overview of literature on sequencing situations with controllable processing times is given in Nowicki and Zdrzalka [11].

In this paper we consider cooperative games arising from sequencing situations with controllable processing times. We show that these games are balanced by obtaining two core elements that depend only on the optimal schedule for the grand coalition. Furthermore we show that many marginal vectors are core elements, in spite of the fact that these games are not convex in general. We also consider convexity for some special cases.

In Section 2 we recall notions from cooperative game theory. Furthermore we formally introduce sequencing situations with controllable processing times and the cooperative games that arise from these situations. In Section 3 we focus on the core of these sequencing games and in Section 4 we consider convexity for some special cases.

## 2 Preliminaries

In this section we will recall some basic notions of cooperative game theory. Then we formally introduce sequencing situations and games with controllable processing times.

### 2.1 Cooperative game theory

A *cooperative game* is a pair  $(N, v)$  where  $N$  is a finite (player-)set and  $v$ , the characteristic function, is a map  $v : 2^N \rightarrow \mathbb{R}$  with  $v(\emptyset) = 0$ . The map  $v$  assigns to each subset  $S \subseteq N$ , called a coalition, a real number  $v(S)$  called the worth of  $S$ . The *core* of a game  $(N, v)$  is the set

$$C(v) = \{x \in \mathbb{R}^N \mid x(S) \geq v(S) \text{ for every } S \subseteq N, x(N) = v(N)\},$$

where  $x(S) = \sum_{i \in S} x_i$ . Intuitively the core is the set of payoff vectors for which no coalition has an incentive to split off from the grand coalition. The core can be empty. If the core of a game is nonempty, the game is called *balanced*. A cooperative game  $(N, v)$  is called *superadditive* if

$$v(S) + v(T) \leq v(S \cup T) \text{ for all } S, T \subseteq N \text{ with } S \cap T = \emptyset,$$

and *convex* if for all  $i, j \in N$  and  $S \subseteq N \setminus \{i, j\}$  it holds

$$v(S \cup \{i\}) + v(S \cup \{j\}) \leq v(S \cup \{i, j\}) + v(S). \quad (1)$$

For convex games the marginal contribution of a player to a coalition exceeds its marginal contribution to a smaller coalition. Note that convex games are balanced ([12]). Now we will introduce marginal vectors. Consider  $\sigma : \{1, \dots, n\} \rightarrow N$ , which describes an order on the player set. Define  $[\sigma(i), \sigma] = \{\sigma(j) : j \leq i\}$  for each  $i \in \{1, \dots, n\}$ . That is,  $[\sigma(i), \sigma]$  is the set of players predecesing player  $\sigma(i)$  with respect to  $\sigma$ . The *marginal vector*  $m^\sigma(v)$  is defined by

$$m_{\sigma(i)}^\sigma(v) = v([\sigma(i), \sigma]) - v([\sigma(i), \sigma] \setminus \{\sigma(i)\}) \text{ for each } i \in \{1, \dots, n\}.$$

That is, each player receives his marginal contribution to the coalition this player joins. We note that a game is convex if and only if all marginal vectors are core elements (cf. [12] and [10]).

A concept that is closely related to convexity and to marginal vectors is permutationally convexity (cf. [6]). An order  $\sigma : \{1, \dots, n\} \rightarrow N$  is a *permutationally convex order* if for all  $0 \leq i \leq k$  and  $S \subseteq N \setminus [\sigma(k), \sigma]$  it holds that

$$v([\sigma(i), \sigma] \cup S) + v([\sigma(k), \sigma]) \leq v([\sigma(k), \sigma] \cup S) + v([\sigma(i), \sigma]). \quad (2)$$

Here we abuse our notation by defining  $[\sigma(0), \sigma] = \emptyset$ . Granot and Huberman [6] show that if  $\sigma : \{1, \dots, n\} \rightarrow N$  is a permutationally convex order, then the corresponding marginal vector  $m^\sigma(v)$  is a core element. The converse of this statement is not true in general.

We end this subsection with the definition of connected coalitions. Let  $\sigma : \{1, \dots, n\} \rightarrow N$  be an order on the player set. A coalition  $S \subseteq N$  is called *connected* with respect to  $\sigma$  if for every  $i, j \in S$  with  $\sigma^{-1}(i) < \sigma^{-1}(j)$  it holds that  $k \in S$  for every  $k \in N$  with  $\sigma^{-1}(i) \leq \sigma^{-1}(k) \leq \sigma^{-1}(j)$ . We use the notation  $\sigma^{-1}$  to denote the inverse of  $\sigma$ , i.e.  $\sigma^{-1}(k) = l$  if and only if  $\sigma(l) = k$ . If a coalition is not connected, then it consists of several *connected components*. For  $T \subseteq N$ , the connected components with respect to  $\sigma$  are denoted by  $T \setminus \sigma$ .

## 2.2 Sequencing situations with controllable processing times

In a sequencing situation with controllable processing times, or cps situation for short, there is a queue of agents, each with one job, in front of the machine. Each agent has to process his job on the machine. The set of agents is denoted by  $N = \{1, \dots, n\}$ . We assume there is an initial processing order on the jobs denoted by  $\sigma_0 : \{1, \dots, n\} \rightarrow N$ . For notational simplicity we assume throughout the paper that  $\sigma_0(i) = i$  for all  $i \in \{1, \dots, n\}$ . The job of agent  $i$ ,  $i \in N$ , has an *initial processing time*  $p_i \geq 0$ . This initial processing time can be reduced to at most  $\bar{p}_i$ , the *crashed processing time* of job  $i$ . The amount of time by which the initial processing time is reduced is the *crash time*. We assume that  $0 \leq \bar{p}_i \leq p_i$  for all  $i \in N$ . The cost of each job is linear in the completion time as well as in the crash time. That is, if  $t$  is the completion time and  $x$  the crash time of job  $i$ , then

$$c_i(t, x) = \alpha_i t + \beta_i x$$

where  $\alpha_i$  and  $\beta_i$  are positive constants expressing the cost of one time unit waiting in the queue and one time unit crashing job  $i$ , respectively. Since crashing a job requires additional resources, we assume that  $\alpha_i \leq \beta_i$  for all  $i \in N$ . That is, the reduction of the processing time of a job by one time unit costs more than the processing of that job by one time unit. A cps situation is now formally defined by the 6-tuple  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$ .

Since the processing times are not fixed, a processing schedule consists of a pair  $(\sigma, x)$  where  $\sigma : \{1, \dots, n\} \rightarrow N$  denotes the processing order, and  $x$  the vector of processing times. The cost of agent  $i \in N$  at processing schedule  $(\sigma, x)$  is now equal to

$$C_i((\sigma, x)) = \alpha_i \left( \sum_{j \in \{1, \dots, n\} : j \leq \sigma^{-1}(i)} x_{\sigma(j)} \right) + \beta_i (p_i - x_i),$$

since  $\sum_{j \in \{1, \dots, n\} : j \leq \sigma^{-1}(i)} x_{\sigma(j)}$  is the completion time of job  $i$  with respect to the schedule  $(\sigma, x)$ , and  $p_i - x_i$  equals the crash time of job  $i$ .

Finding an optimal schedule for a cps situation falls into the class of NP-hard problems. The difficulty of this problem lies in finding the optimal processing times. Once the optimal processing times are known, then it is straightforward to find the optimal processing order by using the Smith-rule ([13]). The Smith-rule states that the jobs should be processed in order of decreasing urgency index, where the urgency index  $u_i$  of a job  $i \in N$  is the quotient between the the completion time cost coefficient and the processing time  $x_i$ , i.e.  $u_i = \frac{\alpha_i}{x_i}$ .

Although finding the optimal processing schedule is difficult, the following lemma, due to Vickson [16], makes the problem a little easier. This lemma states that there is an optimal schedule such that the processing time of each job is either equal to its initial processing time, or its crashed processing time. We note that this result easily follows from the linearity of the cost function.

**Lemma 2.1** ([16]) Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation. There exists an optimal schedule  $(\sigma, x)$  such that  $x_i \in \{p_i, \bar{p}_i\}$  for all  $i \in N$ .

From Lemma 2.1 it follows that the optimal processing schedule can be found by considering all  $2^n$  possibilities for the processing times and applying the Smith-rule for each of these possibilities. Without loss of generality we will assume throughout the paper that optimal schedules satisfy the property of Lemma 2.1, i.e. if  $(\sigma, x)$  is an optimal schedule, then it holds that  $x_i \in \{p_i, \bar{p}_i\}$  for all  $i \in N$ .

### 2.3 Sequencing games with controllable processing times

In this subsection we define sequencing games arising from cps situations, or cps-games for short. Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation. The characteristic function of our game will express the cost savings that each coalition can obtain. For this we have to agree upon which schedules are admissible for a coalition. We will call a processing schedule admissible for a coalition if it satisfies three properties. First, the processing times of players outside the coalition should remain unchanged. Second, the processing times of the players belonging to the coalition should be feasible, i.e. in between the crashed processing time and the initial processing time. And finally, the schedule should be such that the jobs outside the coalition remain in their initial position and no jumps take place over players outside the coalition. Let  $A^S$  denote the set of admissible schedules for coalition  $S \subseteq N$ . Mathematically,  $(\sigma, x) \in A^S$  if it holds that

$$(A1) \quad x_i = p_i \text{ for all } i \in N \setminus S,$$

$$(A2) \quad x_i \in [\bar{p}_i, p_i] \text{ for all } i \in S,$$

$$(A3) \quad \{j \in N : \sigma^{-1}(j) \leq \sigma^{-1}(i)\} = \{j \in N : \sigma_0^{-1}(j) \leq \sigma_0^{-1}(i)\} \text{ for all } i \in N \setminus S.$$

Note that (A3) is the admissibility requirement from Curiel et al. [4]. For almost all sequencing games this admissibility requirement is used. Now we define the cps game  $(N, v)$  by

$$v(S) = \sum_{i \in S} C_i((\sigma_0, p)) - \min_{(\sigma, x) \in A^S} \sum_{i \in S} C_i((\sigma, x)).$$

That is, the worth of each coalition is the maximum cost savings they can obtain by means of an admissible processing schedule. It can easily be seen that cps games are superadditive.

Similarly to Lemma 2.1 it is straightforward to see that for each coalition  $S \subseteq N$ , there is an optimal schedule such that the processing time of each job is either equal to its initial processing time, or to its crashed processing time. Therefore we assume throughout the paper that optimal schedules satisfy this property. We now give an example of a cps game.

**Example 2.1** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be given by  $N = \{1, 2, 3, 4\}$ ,  $\alpha = (1, 1, 1, 1)$ ,  $\beta = (2, 2, 2, 2)$ ,  $p = (10, 4, 3, 15)$ ,  $\bar{p} = (4, 3, 2, 5)$ . Now consider for instance coalition  $\{1, 2, 3\}$ . The optimal schedule for this coalition is given by  $(\sigma, x)$  with  $\sigma = 3214$  and  $x = (10, 4, 2, 15)$ . This yields cost savings  $v(\{1, 2, 3\}) = 41 - 26 = 15$ . The cps game  $(N, v)$  is given by  $v(\{i\}) = 0$  for all  $i \in N$  and

$S$	$\{1, 2\}$	$\{1, 3\}$	$\{1, 4\}$	$\{2, 3\}$	$\{2, 4\}$	$\{3, 4\}$	$\{1, 2, 3\}$	$\{1, 2, 4\}$	$\{1, 3, 4\}$	$\{2, 3, 4\}$	$N$
$v(S)$	6	0	0	1	0	0	15	7	6	2	17

Table 1: A cps game.

From Example 2.1 it immediately follows that cps games are in general not  $\sigma_0$ -component additive games (cf. Curiel et al. [5]). A game is called  $\sigma_0$ -component additive if it is superadditive, and if the worth of each disconnected coalition with respect to  $\sigma_0$  is equal to the sum of the worths of its connected components. In the cps game of Example 2.1 it holds that  $\{1, 3, 4\}$  is a disconnected coalition with respect to  $\sigma_0$ . It consists of connected components  $\{1\}$  and  $\{3, 4\}$ . Since it holds that  $v(\{1\}) + v(\{3, 4\}) < v(\{1, 3, 4\})$  we conclude that the worth of  $\{1, 3, 4\}$  is larger than the sum of the worths of its connected components. Therefore the game is not  $\sigma_0$ -component additive. So in spite of the fact that we use a similar admissibility requirement as Curiel et al. [4] we lose

$\sigma_0$ -component additivity. Note that many sequencing games are proven to be balanced by means of  $\sigma_0$ -component additivity (e.g. Borm et al. [2] and Hamers et al. [7]). Example 2.1 also shows that cps games need not be convex, contrary to the sequencing games of Curiel et al. [4].

**Example 2.2** Consider the cps situation from Example 2.1, and the corresponding game. Consider coalition  $S = \{1, 3\}$ ,  $i = 2$  and  $j = 4$ . Since it holds that  $v(\{1, 2, 3\}) + v(\{1, 3, 4\}) = 21 > 17 = v(N) + v(\{1, 3\})$  we conclude that  $(N, v)$  is not convex.

### 3 The core of cps games

In this section we prove balancedness of cps games by showing the existence of core elements. In particular, we provide two core elements that depend only on the optimal schedule for the grand coalition. Furthermore we show that many marginal vectors are core elements.

In the first part of this section we will provide two core elements that depend on the optimal schedule for the grand coalition. For the perception of these two core elements it is important to note that the optimal processing schedule for the grand coalition can be reached from the initial processing schedule in several ways. For example one could first crash the jobs and then rearrange them. But one could also first rearrange the jobs and then crash them. We emphasize these two possibilities since the construction of our two core elements depends on them.

Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation. Let  $(\sigma, x) \in A^N$  be an optimal processing schedule for the grand coalition. For our first core element, denoted by  $\gamma((\sigma, x))$ , we will reach this optimal processing schedule by first crashing jobs, and then rearranging them. Let  $\gamma((\sigma, x))$  be obtained as follows. First give the cost savings (or costs) obtained by the crashing of a job to the job that crashes and secondly give the cost savings obtained by interchanging two jobs to the back job. Or to put it in a formula:

$$\gamma_i((\sigma, x)) = ((\sum_{j \in N: j \geq i} \alpha_j) - \beta_i)(p_i - x_i) + \sum_{j \in N: j < i} (\alpha_i x_j - \alpha_j x_i)_+ \quad \text{for all } i \in N$$

where  $((\sum_{j \in N: j \geq i} \alpha_j) - \beta_i)(p_i - x_i)$  are the cost savings obtained by crashing job  $i$  by  $(p_i - x_i)$  time. Also observe that if agent  $j$  is directly in front of agent  $i$ , then the cost savings obtainable for  $j$  and  $i$  by switching their order equal  $(\alpha_i x_j - \alpha_j x_i)_+ = \max\{(\alpha_i x_j - \alpha_j x_i), 0\}$ .

For our second core element, denoted by  $\delta((\sigma, x))$ , we reach the optimal schedule  $(\sigma, x)$  by first interchanging jobs to the optimal order, and then crashing them. Let  $\delta((\sigma, x))$  be the following allocation. First give the possibly negative profit of each neighbourswitch to the back job, and secondly give the profit of each crashing to the job that crashes. That is,

$$\delta_i((\sigma, x)) = \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} (\alpha_i p_j - \alpha_j p_i) + ((\sum_{k \in N: \sigma^{-1}(k) \geq \sigma^{-1}(i)} \alpha_k) - \beta_i)(p_i - x_i) \quad \text{for all } i \in N.$$

Because of notational convenience we will write  $\gamma$  and  $\delta$  instead of  $\gamma((\sigma, x))$  and  $\delta((\sigma, x))$  if there can be no confusion about the optimal order that is used. We now give an illustration of  $\gamma$  and  $\delta$ .

**Example 3.1** Let  $N = \{1, 2, 3\}$ ,  $\alpha = (2, 2, 2)$ ,  $\beta = (3, 3, 3)$ ,  $p = (4, 4, 8)$  and  $\bar{p} = (2, 1, 6)$ . The optimal order for the grand coalition is now given by  $\sigma = 213$  and the optimal processing times are  $x = (2, 1, 8)$ . Observe that  $\gamma_3 = \delta_3 = 0$ . Also it holds that  $\gamma_1 = (\alpha_1 + \alpha_2 + \alpha_3 - \beta_1)(p_1 - x_1) = (6 - 3)(4 - 2) = 6$  and  $\gamma_2 = (\alpha_2 + \alpha_3 - \beta_2)(p_2 - x_2) + (\alpha_2 x_1 - \alpha_1 x_2)_+ = (4 - 3)(4 - 1) + (4 - 2) = 5$ . Furthermore, note that  $\delta_1 = (\alpha_1 + \alpha_3 - \beta_1)(p_1 - x_1) = (4 - 3)(4 - 2) = 2$  and that  $\delta_2 = (\alpha_2 p_1 - \alpha_1 p_2) + (\alpha_2 + \alpha_1 + \alpha_3 - \beta_2)(p_2 - x_2) = (8 - 8) + (6 - 3)(4 - 1) = 9$ .

Before we show that  $\gamma$  and  $\delta$  are core elements, we need two lemmas. In the first we obtain another expression for  $\delta$  which we will use in the proof of Theorem 3.1. The proof of this lemma can be found in the Appendix. The second lemma is a technical but straightforward lemma which we will use throughout the paper.

**Lemma 3.1** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation and let  $(\sigma, x) \in A^N$  be an optimal processing schedule. For all  $i \in N$  it holds that

$$\delta_i = \gamma_i - \sum_{j \in N: j > i \text{ and } \sigma^{-1}(j) < \sigma^{-1}(i)} \alpha_j(p_i - x_i) + \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} \alpha_i(p_j - x_j).$$

**Lemma 3.2** Let  $a_1, a_2, q_1, \bar{q}_1, q_2 \geq 0$  with  $q_1 \geq \bar{q}_1$ . It holds that  $a_2(q_1 - \bar{q}_1) + (a_2\bar{q}_1 - a_1q_2)_+ \geq (a_2q_1 - a_1q_2)_+$ .

**Proof:** If  $(a_2q_1 - a_1q_2)_+ = 0$  then the inequality is trivially satisfied, so suppose that  $(a_2q_1 - a_1q_2)_+ > 0$ . This implies that  $a_2q_1 - a_1q_2 > 0$ . Straightforwardly it follows that

$$\begin{aligned} a_2(q_1 - \bar{q}_1) + (a_2\bar{q}_1 - a_1q_2)_+ &\geq a_2(q_1 - \bar{q}_1) + (a_2\bar{q}_1 - a_1q_2) \\ &= a_2q_1 - a_1q_2 = (a_2q_1 - a_1q_2)_+. \end{aligned}$$

□

Now we are ready to prove that  $\gamma \in C(v)$  and  $\delta \in C(v)$ .

**Theorem 3.1** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation and let  $(N, v)$  be the corresponding game. Let  $(\sigma, x) \in A^N$  be an optimal processing schedule. It holds that  $\gamma \in C(v)$  and  $\delta \in C(v)$ .

**Proof:** It is obvious that  $\gamma$  and  $\delta$  are efficient, since both  $\gamma$  and  $\delta$  divide the total cost savings of the grand coalition. Let  $T \subseteq N$ . We need to show that  $\sum_{i \in T} \gamma_i \geq v(T)$  and  $\sum_{i \in T} \delta_i \geq v(T)$ . We will equivalently show that  $\sum_{i \in N \setminus T} \gamma_i + v(T) \leq v(N)$  and  $\sum_{i \in N \setminus T} \delta_i + v(T) \leq v(N)$  by constructing a suboptimal schedule  $(\sigma^{subopt}, p^{subopt}) \in A^N$  that depends on an optimal processing schedule of  $T$ . We show that the total cost savings obtained in this suboptimal schedule exceed both  $\sum_{i \in N \setminus T} \gamma_i + v(T)$  and  $\sum_{i \in N \setminus T} \delta_i + v(T)$ . Obviously the cost savings in the suboptimal schedule are at most  $v(N)$ .

We first find expressions for  $\sum_{i \in N \setminus T} \gamma_i$  and  $\sum_{i \in N \setminus T} \delta_i$ . Note that

$$\begin{aligned} &\sum_{i \in N \setminus T} \gamma_i \\ &= \sum_{i \in N \setminus T} ((\sum_{j \in N: j \geq i} \alpha_j) - \beta_i)(p_i - x_i) \end{aligned} \tag{3}$$

$$+ \sum_{i, j \in N \setminus T: j < i} (\alpha_i x_j - \alpha_j x_i) \tag{4}$$

$$+ \sum_{j \in T, i \in N \setminus T: j < i} (\alpha_i x_j - \alpha_j x_i) \tag{5}$$

and that

$$\begin{aligned} &\sum_{i \in N \setminus T} \delta_i \\ &= \sum_{i \in N \setminus T} \gamma_i \\ &- \sum_{i \in N \setminus T} \sum_{j \in N: j > i \text{ and } \sigma^{-1}(j) < \sigma^{-1}(i)} \alpha_j(p_i - x_i) \end{aligned} \tag{6}$$

$$+ \sum_{i \in N \setminus T} \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} \alpha_i(p_j - x_j), \tag{7}$$

$$= \sum_{i \in N \setminus T} \gamma_i \quad (8)$$

$$- \sum_{i \in N \setminus T} \sum_{j \in T: j > i \text{ and } \sigma^{-1}(j) < \sigma^{-1}(i)} \alpha_j (p_i - x_i) \quad (9)$$

$$+ \sum_{i \in N \setminus T} \sum_{j \in T: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} \alpha_i (p_j - x_j), \quad (10)$$

where the first equality holds because of Lemma 3.1. The second equality is satisfied because for each pair  $i, j \in N \setminus T$  with  $j > i$  and  $\sigma^{-1}(j) < \sigma^{-1}(i)$  every term in (6) also appears in (7) but with the opposite sign.

Let  $(\sigma^T, p^T) \in A^T$  be an optimal schedule for coalition  $T$ . Consider the suboptimal schedule  $(\sigma^{subopt}, p^{subopt}) \in A^N$ , where  $p_j^{subopt} = p_j^T$  if  $j \in T$ ,  $p_j^{subopt} = x_j$  if  $j \in N \setminus T$ , and  $\sigma^{subopt}$  is the order obtained by applying the Smith-rule using the suboptimal processing times. The total cost savings for the grand coalition at this suboptimal schedule equal

$$P = \sum_{j \in T} ((\sum_{i \in T: i \geq j} \alpha_i) - \beta_j) (p_j - p_j^{subopt}) \quad (11)$$

$$+ \sum_{j \in T} ((\sum_{i \in N \setminus T: i > j} \alpha_i) (p_j - p_j^{subopt}) \quad (12)$$

$$+ \sum_{i \in N \setminus T} ((\sum_{j \in N: j \geq i} \alpha_j) - \beta_i) (p_i - p_i^{subopt}) \quad (13)$$

$$+ \sum_{j, i \in N \setminus T: j < i} (\alpha_i p_j^{subopt} - \alpha_j p_i^{subopt})_+ \quad (14)$$

$$+ \sum_{j \in N \setminus T, i \in T: j < i} (\alpha_i p_j^{subopt} - \alpha_j p_i^{subopt})_+ \quad (15)$$

$$+ \sum_{j \in T, i \in N \setminus T: j < i} (\alpha_i p_j^{subopt} - \alpha_j p_i^{subopt})_+ \quad (16)$$

$$+ \sum_{j, i \in T: j < i} (\alpha_i p_j^{subopt} - \alpha_j p_i^{subopt})_+. \quad (17)$$

Expressions (11) and (12) are the cost savings obtained by crashing the jobs of  $T$ , and expression (13) the cost savings obtained by crashing the jobs of  $N \setminus T$ . The cost savings obtained by rearranging the jobs are equal to the sum of expressions (14), (15), (16) and (17).

Now we will show that the sum of (3), (4), (5), (10) and  $v(T)$  is exceeded by  $P$ . Since (10) is nonnegative, this shows that  $\sum_{i \in N \setminus T} \gamma_i + v(T)$  is exceeded by  $P$ . Furthermore, since (9) is nonpositive, it also shows that  $\sum_{i \in N \setminus T} \delta_i + v(T)$  is exceeded by  $P$ .

First note that the sum of expressions (11) and (17) exceeds  $v(T)$  because  $p_i^{subopt} = p_i^T$  for all  $i \in T$ . It also holds that (13) coincides with (3) as well as (14) coincides with (4) because  $p_j^{subopt} = x_j$  for all  $j \in N \setminus T$ . Finally, note that expression (15) is nonnegative. Hence, for showing that the sum of expressions (3), (4), (5), (10) and  $v(T)$  is exceeded by  $P$  it is sufficient to show that the sum of (5) and (10) is exceeded by the sum of (12) and (16). We will show that this holds by comparing the sums term by term.

So let  $j \in T$ ,  $i \in N \setminus T$  such that  $j < i$ . Note that it now holds that  $p_i^{subopt} = x_i$  and that  $p_j^{subopt} = p_j^T$ . We distinguish between two cases.

**Case 1:**  $\sigma^{-1}(j) \leq \sigma^{-1}(i)$ .

In this case it holds that  $i$  and  $j$  do not have a corresponding term in (10). Therefore we only need to compare the corresponding term in (5) with the corresponding terms in (12) and (16).



If it holds that  $p_j^{subopt} \geq x_j$ , then we have that  $(\alpha_i p_j^{subopt} - \alpha_j p_i^{subopt})_+ = (\alpha_i p_j^{subopt} - \alpha_j x_i)_+ \geq (\alpha_i x_j - \alpha_j x_i)_+$ . Hence, the term in (5) is exceeded by the corresponding term in (16). Since the corresponding term in (12) is nonnegative we conclude that the term in (5) is exceeded by the sum of the corresponding terms in (12) and (16).

So assume that  $p_j^{subopt} < x_j$ . Since optimal processing times only take two values by assumption it holds that  $p_j^{subopt} = \bar{p}_j$  and  $x_j = p_j$ . Since  $j < i$  it follows that there is a term  $\alpha_i(p_j - p_j^{subopt})$  in expression (12). Now according to Lemma 3.2 using  $a_1 = \alpha_j$ ,  $a_2 = \alpha_i$ ,  $q_1 = p_j$ ,  $\bar{q}_1 = p_j^{subopt}$  and  $q_2 = p_i$ , it holds that  $\alpha_i(p_j - p_j^{subopt}) + (\alpha_i p_j^{subopt} - \alpha_j p_i^{subopt})_+ \geq (\alpha_i p_j - \alpha_j p_i^{subopt})_+ = (\alpha_i x_j - \alpha_j x_i)_+$ , where the equality holds because  $p_j = x_j$  and  $p_i^{subopt} = x_i$ .

So in this case it holds that the corresponding terms in (12) and (16) exceed the terms in (5).

**Case 2:**  $\sigma^{-1}(j) > \sigma^{-1}(i)$ .

Since  $\sigma^{-1}(j) > \sigma^{-1}(i)$  it necessarily holds  $\frac{\alpha_i}{x_i} \geq \frac{\alpha_j}{x_j}$  and thus that  $\alpha_i x_j - \alpha_j x_i \geq 0$ . Straightforwardly we obtain that

$$\begin{aligned} & \alpha_i(p_j - p_j^{subopt}) + (\alpha_i p_j^{subopt} - \alpha_j p_i^{subopt})_+ \geq \alpha_i(p_j - p_j^{subopt}) + (\alpha_i p_j^{subopt} - \alpha_j p_i^{subopt}) \\ &= \alpha_i p_j - \alpha_j p_i^{subopt} = \alpha_i(p_j - x_j) + (\alpha_i x_j - \alpha_j p_i^{subopt}) \\ &= \alpha_i(p_j - x_j) + (\alpha_i x_j - \alpha_j x_i)_+, \end{aligned}$$

where the last equality holds since  $p_i^{subopt} = x_i$  and  $\alpha_i x_j - \alpha_j x_i \geq 0$ . We conclude that the corresponding terms of expressions (12) and (16) exceed the corresponding terms of expressions (5) and (10). This completes the proof.  $\square$

Note that if the optimal schedule is not unique, then  $\gamma$  and  $\delta$  are not uniquely determined, since  $\gamma$  and  $\delta$  clearly depend on the optimal schedule that is used. This possibility is illustrated in the following example.

**Example 3.2** Let  $N = \{1, 2\}$ ,  $\alpha = (2, 2)$ ,  $\beta = (3, 3)$ ,  $p = (2, 2)$  and  $\bar{p} = (1, 1)$ . Observe that jobs 1 and 2 are completely symmetric. There are two optimal processing schedules: only crash job 1 and keep the initial processing order, and only crash job 2 and interchange job 1 and 2. Mathematically,  $(\sigma^1, x^1)$  is optimal with  $\sigma^1 = 12$  and  $x^1 = (1, 2)$  and  $(\sigma^2, x^2)$  is optimal with  $\sigma^2 = 21$  and  $x^2 = (2, 1)$ . It holds that  $\gamma_1((\sigma^1, x^1)) = \delta_1((\sigma^1, x^1)) = v(\{1, 2\}) = 1$  and  $\gamma_2((\sigma^1, x^1)) = \delta_2((\sigma^1, x^1)) = 0$ . Furthermore it holds that  $\gamma_1((\sigma^2, x^2)) = \delta_1((\sigma^2, x^2)) = 0$  and  $\gamma_2((\sigma^2, x^2)) = \delta_2((\sigma^2, x^2)) = v(\{1, 2\}) = 1$ . According to Theorem 3.1 all four allocations are core elements.

In the upcoming part of this section we study marginal vectors that provide core elements. In the next theorem we show that many marginal vectors are core elements by showing that the corresponding orders are permutationally convex orders. In particular we show that the orders that put the jobs in reverse order, with respect to  $\sigma_0$ , but have job 1 at an arbitrary position are permutationally convex.

**Theorem 3.2** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation and let  $(N, v)$  be the corresponding game. Let  $1 \leq j \leq n$  and let  $\sigma : \{1, \dots, n\} \rightarrow N$  be such that  $\sigma(i) = n + 1 - i$  for all  $1 \leq i < j$ ,  $\sigma(i) = n + 2 - i$  for all  $j < i \leq n$  and  $\sigma(j) = 1$ . Then it holds that  $\sigma$  is a permutationally convex order. In particular it holds that  $m^\sigma(v) \in C(v)$ .

Since the proof is rather cumbersome, we have put it in the Appendix. The following example illustrates Theorem 3.2.

**Example 3.3** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be as in Example 2.1 and let  $(N, v)$  be the corresponding game. According to Theorem 3.2 the marginal vectors corresponding to the orders 4321, 4312, 4132 and 1432 are all core elements. In particular,  $m^{4321}(v) = (15, 2, 0, 0)$ ,  $m^{4312}(v) = (6, 11, 0, 0)$  and  $m^{4132}(v) = m^{1432}(v) = (0, 11, 6, 0) \in C(v)$ .

The orders in Theorem 3.2 are not the only permutationally convex orders of cps games. In Van Velzen [15] it is shown that for any TU-game  $(N, v)$  it holds that if  $\sigma : \{1, \dots, n\} \rightarrow N$  is a permutationally convex order, then  $\sigma_{n-1}$  is a permutationally convex order as well, where  $\sigma_{n-1}$  is obtained by interchanging the players in the  $(n-1)$ -st and  $n$ -th position of  $\sigma$ . In the proceeding we will show for cps games that if  $\sigma : \{1, \dots, n\} \rightarrow N$  is a permutationally convex order, then  $\sigma_1$  is a permutationally convex order as well. That is, the order obtained from  $\sigma$  by interchanging the first and second position is also permutationally convex. We first need the following lemma.

**Lemma 3.3** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a sequencing situation with controllable processing times and let  $(N, v)$  be the corresponding game. Let  $S, T \subseteq N$  with  $|S \cap T| = 1$ . Then it holds that  $v(S) + v(T) \leq v(S \cup T)$ .

The proof of this lemma is also recorded in the Appendix. Superadditivity of cps games together with Lemma 3.3 implies the following corollary.

**Corollary 3.1** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation with  $|N| = 3$ . Let  $(N, v)$  be the corresponding game. Then  $(N, v)$  is convex.

As promised, we now show that if  $\sigma : \{1, \dots, n\} \rightarrow N$  is a permutationally convex order, then the order obtained from  $\sigma$  by interchanging the players in the first and second position,  $\sigma_1$ , is a permutationally convex order as well.

**Theorem 3.3** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation and let  $(N, v)$  be the corresponding game. If  $\sigma : \{1, \dots, n\} \rightarrow N$  is a permutationally convex order, then  $\sigma_1$  is a permutationally convex order as well. In particular it holds that  $m^{\sigma_1} \in C(v)$ .

**Proof:** Suppose that  $\sigma : \{1, \dots, n\} \rightarrow N$  is a permutationally convex order. We need to show that for all  $0 \leq i \leq k$  and all  $S \subseteq N \setminus [\sigma_1(k), \sigma_1]$  it holds that

$$v([\sigma_1(i), \sigma_1] \cup S) + v([\sigma_1(k), \sigma_1]) \leq v([\sigma_1(k), \sigma_1] \cup S) + v([\sigma_1(i), \sigma_1]).$$

If  $i = 0$ , then the inequality is trivially satisfied because  $(N, v)$  satisfies superadditivity. If  $i \geq 2$ , then this inequality is satisfied since in that case  $[\sigma_1(i), \sigma_1] = [\sigma(i), \sigma]$ ,  $[\sigma_1(k), \sigma_1] = [\sigma(k), \sigma]$  and our assumption that  $\sigma$  is a permutationally convex order. So let  $i = 1$  and  $k > i$ . Since  $([\sigma_1(i), \sigma_1] \cup S) \cap [\sigma_1(k), \sigma_1] = \{\sigma_1(1)\}$ , the inequality holds by Lemma 3.3.  $\square$

We now illustrate the result of Van Velzen [15] about the permutational convexity of  $\sigma_{n-1}$  and Theorem 3.3.

**Example 3.4** Let  $(N, v)$  be as in Example 2.1. In Example 3.3 we showed that 4321, 4312, 4132 and 1432 are permutationally convex orders. From Van Velzen [15] and Theorem 3.3 it follows that 3421, 3412, 4123 and 1423 are permutationally convex as well. Thus,  $m^{3421}(v) = (15, 2, 0, 0)$ ,  $m^{3412}(v) = (6, 11, 0, 0)$  and  $m^{4123}(v) = m^{1423}(v) = (0, 7, 10, 0)$  are all core elements.

The final theorem of this section shows a way to alter a core element slightly, such that the new allocation is still in the core.

**Theorem 3.4** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation and  $(N, v)$  be the corresponding game. Let  $x \in C(v)$  and let  $j, k \in N$ . Furthermore let  $\lambda \geq 0$  be such that  $\lambda \leq (v(\{j, k\}) - x_k)_+$  and let  $\bar{x}$  be such that  $\bar{x}_i = x_i$  for all  $i \in N \setminus \{j, k\}$ ,  $\bar{x}_j = x_j - \lambda$  and  $\bar{x}_k = x_k + \lambda$ . Then it holds that  $\bar{x} \in C(v)$ .

**Proof:** If  $\lambda = 0$ , then it follows that  $\bar{x} = x$ , and trivially we have that  $\bar{x} \in C(v)$ . So suppose that  $\lambda > 0$ . It now follows by definition of  $\lambda$  that  $x_k + \lambda \leq v(\{j, k\})$ .

Showing that  $\bar{x} \in C(v)$  boils down to showing that for each  $S \subseteq N \setminus \{k\}$  with  $j \in S$  it holds that  $\bar{x}(S) \geq v(S)$ . Let  $S \subseteq N \setminus \{k\}$  such that  $j \in S$ . Now note that

$$\sum_{i \in S \cup \{k\}} x_i \geq v(S \cup \{k\}) \geq v(S) + v(\{j, k\}), \quad (18)$$

where the first inequality holds because  $x \in C(v)$  and the second follows by Lemma 3.3. Thus

$$\sum_{i \in S} \bar{x}_i = \sum_{i \in S} x_i - \lambda = \sum_{i \in S \cup \{k\}} x_i - x_k - \lambda \geq v(S) + v(\{j, k\}) - x_k - \lambda \geq v(S),$$

where the first inequality follows by expression (18), and the second because  $x_k + \lambda \leq v(\{j, k\})$ .  $\square$

Theorem 3.4 enables us to show a nice feature of cps games. Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation and let  $(N, v)$  be the corresponding game. Let  $\sigma : \{1, \dots, n\} \rightarrow N$  be such that  $m^\sigma(v) \in C(v)$ . If  $k$  and  $j$  are the first and second player according to  $\sigma$ , respectively, then it holds that  $m_k^\sigma(v) = 0$  and  $m_j^\sigma(v) = v(\{j, k\})$ . Let  $\lambda = v(\{j, k\})$ . According to Theorem 3.4 it holds that  $\bar{x} \in C(v)$ , where  $\bar{x}$  is given by  $\bar{x}_k = m_k^\sigma(v) + \lambda = v(\{j, k\})$ ,  $\bar{x}_j = m_j^\sigma(v) - \lambda = 0$  and  $\bar{x}_i = m_i^\sigma(v)$  for all  $i \in N \setminus \{j, k\}$ . Now note that  $\bar{x} = m^{\sigma^1}(v)$ , and thus that  $m^{\sigma^1}(v) \in C(v)$ . Therefore we have the following proposition.

**Proposition 3.1** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation and let  $(N, v)$  be the corresponding game. If  $\sigma : \{1, \dots, n\} \rightarrow N$  is such that  $m^\sigma(v) \in C(v)$ , then  $m^{\sigma^1}(v) \in C(v)$ . In particular, the number of marginal vectors in the core is even.

## 4 Convexity of cps games

In this section we investigate convexity of cps games. In particular we show for situations where all completion time cost coefficients, all crash time cost coefficients and all maximum crash times are equal, then the corresponding game is convex. Furthermore we show that relaxing this condition might lead to nonconvex cps games.

The next theorem shows that a specific class of cps situations has corresponding convex games.

**Theorem 4.1** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation and let  $(N, v)$  be the corresponding game. If  $\alpha_i = \alpha_j$ ,  $\beta_i = \beta_j$  and  $p_i - \bar{p}_i = p_j - \bar{p}_j$  for all  $i, j \in N$ , then  $(N, v)$  is convex.

**Proof:** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation with  $\alpha_i = \alpha_j$ ,  $\beta_i = \beta_j$  and  $p_i - \bar{p}_i = p_j - \bar{p}_j$  for all  $i, j \in N$  and let  $(N, v)$  be the corresponding game. For notational convenience we say that  $\alpha_i = \alpha$ ,  $\beta_i = \beta$  and  $p_i - \bar{p}_i = q$  for all  $i \in N$ , where  $\alpha$ ,  $\beta$  and  $q$  here denote scalars and not vectors. We assume that  $q > 0$ . If  $q = 0$  then no crashing is possible and the cps situation is nothing more than a standard sequencing situation. The resulting game will be convex by Curiel et al. [4].

First we show that the optimal schedule is easy to determine in this case. Let  $S \subseteq N$ . We will reach the optimal schedule for  $S$  by first rearranging the jobs according to the Smith-rule and then crashing them. Note that, since  $\alpha_i = \alpha$  and  $\beta_i = \beta$  for all  $i \in S$ , it is easy to determine the position of the jobs that are crashed in the optimal schedule. In particular, if  $k \in S$  is the  $l$ -th job of  $S$  in the optimal processing order, then there are  $|S| - l$  jobs of  $S$  in the queue behind  $k$ . Hence, if job

$k$  crashes, then this yields cost savings of  $((|S| - l + 1)\alpha - \beta)q$ . This term is nonnegative only if  $(|S| - l + 1)\alpha - \beta$  is nonnegative. Furthermore, observe that these cost savings do not depend on  $k$ . We conclude that the cost savings due to crashing jobs will equal  $\sum_{k=1}^{|S|} ((|S| - k + 1)\alpha - \beta)_+ q$ . These cost savings only depend on the size of  $S$  and not on the jobs of  $S$  or on the order in which the jobs are processed.

Because the cost savings obtained from optimally crashing jobs are independent of the order in which the jobs are processed it follows that the total cost savings are maximized if the cost savings obtained from interchanging the jobs are maximized. In particular, the total cost savings are maximized if the jobs are lined up in order of decreasing urgencies. Therefore

$$v(S) = \sum_{T \in S \setminus \sigma_0} \sum_{i, j \in T: i < j} (\alpha_j p_i - \alpha_i p_j)_+ + \sum_{k=1}^{|S|} ((|S| - k + 1)\alpha - \beta)_+ q.$$

Since  $(N, z)$  with  $z(S) = \sum_{T \in S \setminus \sigma_0} \sum_{i, j \in T: i < j} (\alpha_j p_i - \alpha_i p_j)_+$  is convex (Curiel et al. [4]), it is sufficient for convexity of  $(N, v)$  to show that  $(N, w)$  with  $w(S) = \sum_{k=1}^{|S|} ((|S| - k + 1)\alpha - \beta)_+ q$  is convex. So let  $i, j \in N$  and  $S \subseteq N \setminus \{i, j\}$ . We distinguish between three cases in order to show that  $w(S \cup \{i\}) + w(S \cup \{j\}) \leq w(S \cup \{i, j\}) + w(S)$ .

**Case 1:**  $w(S \cup \{i\}) = w(S \cup \{j\}) = 0$ .

Trivially it follows that  $w(S \cup \{i\}) + w(S \cup \{j\}) = 0 \leq w(S \cup \{i, j\}) + w(S)$ .

**Case 2:**  $w(S \cup \{i\}) = w(S \cup \{j\}) > 0$  and  $w(S) = 0$ .

Since  $w(S) = 0$  and  $q > 0$ , it holds that  $|S|\alpha - \beta \leq 0$ . Therefore it holds that  $w(S \cup \{i\}) = w(S \cup \{j\}) = ((|S| + 1)\alpha - \beta)q$ . Hence,

$$\begin{aligned} w(S \cup \{i\}) + w(S \cup \{j\}) &= 2((|S| + 1)\alpha - \beta)q \\ &\leq ((|S| + 1)\alpha - \beta)q + ((|S| + 2)\alpha - \beta)q = w(S \cup \{i, j\}) = w(S \cup \{i, j\}) + w(S). \end{aligned}$$

**Case 3:**  $w(S) > 0$ .

Because  $w(S) > 0$ , it holds that  $w(S \cup \{i\}) = w(S) + ((|S| + 1)\alpha - \beta)q$ . Furthermore we have that  $w(S \cup \{i, j\}) = w(S \cup \{j\}) + ((|S| + 2)\alpha - \beta)q$ . Therefore,

$$\begin{aligned} w(S \cup \{i\}) + w(S \cup \{j\}) &= w(S) + ((|S| + 1)\alpha - \beta)q + w(S \cup \{j\}) \\ &\leq w(S) + ((|S| + 2)\alpha - \beta)q + w(S \cup \{j\}) = w(S) + w(S \cup \{i, j\}). \end{aligned} \quad \square$$

Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation with  $\alpha_i = \alpha_j$ ,  $\beta_i = \beta_j$  and  $p_i - \bar{p}_i = p_j - \bar{p}_j$  for all  $i, j \in N$ . Let  $(N, v)$  be the corresponding game. According to the proof of Theorem 4.1 it holds that  $(N, v)$  is the sum of a symmetric game and a classical sequencing game à la Curiel et al. [4]. Hence, the Shapley value of  $(N, v)$  is equal to the sum of the Shapley value of the symmetric game and the Shapley value of the classical sequencing game. Since both can be computed easily it follows that the Shapley value of cps games arising from these special cps situations can be computed easily.

The following examples show that by relaxing the condition of Theorem 4.1 convexity might be lost.

**Example 4.1** ( $\alpha_i = \alpha_j$ ,  $\beta_i = \beta_j$  and  $\bar{p}_i = \bar{p}_j$  for all  $i, j \in N$ ) Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be given by  $N = \{1, 2, 3, 4\}$ ,  $\alpha_i = 2$ ,  $\beta_i = 5$  and  $\bar{p}_i = 2$  for all  $i \in N$ . Furthermore, let  $p = (7, 3, 3, 7)$ . Let  $(N, v)$  be the corresponding game. Since  $2\alpha < \beta$ , it follows for coalition  $\{1, 3\}$  no cost savings can be obtained by crashing job 1. Since also no cost savings can be obtained by rearranging the jobs, it follows that  $v(\{1, 3\}) = 0$ . Because  $2\alpha < \beta$  and  $3\alpha > \beta$  it follows that for the grand coalition it is optimal to crash exactly two jobs, namely the jobs that are in the first and

second position of the optimal processing order. By considering all possibilities of crashing two of the four jobs, it is straightforward to see that the schedule  $(2314, (7, 2, 2, 7)) \in A^N$  is optimal. As a result we have that  $v(N) = 20$ . Now observe that  $(2314, (7, 2, 3, 7)) \in A^{\{1,2,3\}}$  and that  $(1234, (2, 3, 3, 7)) \in A^{\{1,3,4\}}$ . These schedules yield cost savings of 17 and 5 for coalitions  $\{1, 2, 3\}$  and  $\{1, 3, 4\}$  respectively. Therefore  $v(\{1, 2, 3\}) \geq 17$  and  $v(\{1, 3, 4\}) \geq 5$ . We conclude that  $v(\{1, 2, 3\}) + v(\{1, 3, 4\}) \geq 22 > 20 = v(N) + v(\{1, 3\})$ , and thus that  $(N, v)$  is not convex.

**Example 4.2** ( $\alpha_i = \alpha_j$ ,  $p_i = p_j$  and  $\bar{p}_i = \bar{p}_j$  for all  $i, j \in N$ ) Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be given by  $N = \{1, 2, 3, 4, 5\}$ ,  $\alpha_i = 2$ ,  $p_i = 2$  and  $\bar{p}_i = 1$  for all  $i \in N$ . Furthermore, let  $\beta = (6, 6, 3, 3, 6)$ . Let  $(N, v)$  be the corresponding game. For each coalition, the optimal order can be reached by first interchanging jobs, and then crashing them. Since  $\alpha_i = 2$  and  $p_i = 2$  for all  $i \in N$  it follows that  $(\alpha_i p_j - \alpha_j p_i) = 0$  for all  $i, j \in N$ . That is, first rearranging jobs yields no cost savings and no extra costs. Hence, the cost savings for each coalition consist of cost savings due to crashing only. Because  $\alpha_i = 2$  and  $p_i - \bar{p}_i = 1$  for all  $i \in N$ , it is optimal for each coalition to put the jobs with lowest  $\beta_i$  to the front as much as possible. In particular, for coalitions  $\{1, 3, 4\}$  and  $N$  the optimal schedules are  $(12345, (2, 2, 1, 2, 2)) \in A^{\{1,3,4\}}$  and  $(34125, (2, 2, 1, 1, 2)) \in A^N$  respectively. This yields cost savings of  $v(\{1, 3, 4\}) = 1$  and  $v(N) = 12$ . For coalitions  $\{1, 2, 3, 4\}$  and  $\{1, 3, 4, 5\}$  the optimal schedules are given by  $(34125, (2, 2, 1, 1, 2)) \in A^{\{1,2,3,4\}}$  and  $(12345, (1, 2, 1, 1, 2)) \in A^{\{1,3,4,5\}}$ , respectively, with cost savings  $v(\{1, 2, 3, 4\}) = 8$  and  $v(\{1, 3, 4, 5\}) = 6$ . We conclude that  $v(\{1, 2, 3, 4\}) + v(\{1, 3, 4, 5\}) = 14 > 13 = v(N) + v(\{1, 3, 4\})$ , and thus that  $(N, v)$  is not convex.

**Example 4.3** ( $\beta_i = \beta_j$ ,  $p_i = p_j$  and  $\bar{p}_i = \bar{p}_j$  for all  $i, j \in N$ ) Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be given by  $N = \{1, 2, 3, 4\}$ ,  $\beta_i = 5$ ,  $p_i = 2$  and  $\bar{p}_i = 1$  for all  $i \in N$ . Furthermore, let  $\alpha = (1, 1, 5, 1)$ . Let  $(N, v)$  be the corresponding game. Since coalition  $\{1, 3\}$  is a disconnected coalition, the only cost savings it can obtain is by crashing job 1. Hence,  $v(\{1, 3\}) = 1$ . For the grand coalition the schedule  $(3124, (2, 2, 1, 2)) \in A^N$  is optimal, with cost savings  $v(N) = 19$ . Furthermore it holds that  $(3124, (2, 2, 1, 2)) \in A^{\{1,2,3\}}$  and that  $(1234, (1, 2, 1, 2)) \in A^{\{1,3,4\}}$ . These schedule lead to cost savings of 18 and 3 for coalitions  $\{1, 2, 3\}$  and  $\{1, 3, 4\}$  respectively. We conclude that  $v(\{1, 2, 3\}) + v(\{1, 3, 4\}) \geq 21 > 20 = v(N) + v(\{1, 3\})$ , and thus that  $(N, v)$  is not convex.

As a final remark to this paper we conjecture that for a cps situation  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  the corresponding game is convex if it holds that  $\alpha_i = \alpha_j$ ,  $\beta_i = \beta_j$  and  $p_i = p_j$  for all  $i, j \in N$ .

## References

- [1] B. Alidaee and A. Ahmadian. Two parallel machine sequencing problems involving controllable job processing times. *European Journal of Operational Research*, 70:335–341, 1993.
- [2] P. Borm, G. Fiestras-Janeiro, H. Hamers, E. Sanchez, and M. Voorneveld. On the convexity of sequencing games with due dates. *European Journal of Operational Research*, 136:616–634, 2002.
- [3] P. Calleja, P. Borm, H. Hamers, F. Klijn, and M. Slikker. On a new class of parallel sequencing situations and related games. *Annals of Operations Research*, 109:265–277, 2002.
- [4] I. Curiel, G. Pederzoli, and S. Tijs. Sequencing games. *European Journal of Operational Research*, 40:344–351, 1989.
- [5] I. Curiel, J. Potters, V. Rajendra Prasad, S. Tijs, and B. Veltman. Sequencing and cooperation. *Operations Research*, 42:566–568, 1994.
- [6] D. Granot and G. Huberman. The relationship between convex games and minimal cost spanning tree games: A case for permutationally convex games. *SIAM Journal of Algorithms and Discrete Methods*, 3:288–292, 1982.

- [7] H. Hamers, P. Borm, and S. Tijs. On games corresponding to sequencing situations with ready times. *Mathematical Programming*, 70:1–13, 1995.
- [8] H. Hamers, F. Klijn, and J. Suijs. On the balancedness of multimachine sequencing games. *European Journal of Operational Research*, 119:678–691, 1999.
- [9] H. Hamers, F. Klijn, and B. van Velzen. On the convexity of precedence sequencing games. CentER Discussion Paper 2002-112, Tilburg University, Tilburg, The Netherlands, 2002.
- [10] T. Ichiishi. Supermodularity: applications to convex games and the greedy algorithm for lp. *Journal of Economic Theory*, 25:283–286, 1981.
- [11] E. Nowicki and S. Zdrzalka. A survey of results for sequencing problems with controllable processing times. *Discrete Applied Mathematics*, 26:271–287, 1990.
- [12] L. Shapley. Cores of convex games. *International Journal of Game Theory*, 1:11–26, 1971.
- [13] W. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3:59–66, 1956.
- [14] A. van den Nouweland, M. Krabbenborg, and J. Potters. Flowshops with a dominant machine. *European Journal of Operational Research*, 62:38–46, 1992.
- [15] B. van Velzen. A note on permutationally convex games. Mimeo, Tilburg University, Tilburg, The Netherlands, 2003.
- [16] R.G. Vickson. Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine. *Operations Research*, 28(5):1155–1167, 1980.
- [17] R.G. Vickson. Two single machine sequencing problems involving controllable processing times. *AIIE Transactions*, 12(3):258–262, 1980.

## Appendix

**Proof of Lemma 3.1:** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation. Let  $(\sigma, x) \in A^N$  be an optimal schedule for the grand coalition. Let  $i \in N$ . Recall that

$$\begin{aligned} & \delta_i \\ = & \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} (\alpha_i p_j - \alpha_j p_i) \end{aligned} \quad (19)$$

$$+ \left( \left( \sum_{k \in N: \sigma^{-1}(k) \geq \sigma^{-1}(i)} \alpha_k \right) - \beta_i \right) (p_i - x_i) \quad (20)$$

First note for expression (19) that

$$\begin{aligned} & \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} (\alpha_i p_j - \alpha_j p_i) \\ = & \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} (\alpha_i x_j - \alpha_j p_i) \end{aligned} \quad (21)$$

$$+ \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} \alpha_i (p_j - x_j). \quad (22)$$

Furthermore, note for expression (20) that

$$\begin{aligned} & ((\sum_{k \in N: \sigma^{-1}(k) \geq \sigma^{-1}(i)} \alpha_k) - \beta_i)(p_i - x_i) \\ = & ((\sum_{k \in N: k \geq i} \alpha_k) - \beta_i)(p_i - x_i) \end{aligned} \quad (23)$$

$$- \sum_{j \in N: j > i \text{ and } \sigma^{-1}(j) < \sigma^{-1}(i)} \alpha_j(p_i - x_i) \quad (24)$$

$$+ \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} \alpha_j(p_i - x_i). \quad (25)$$

Now adding expressions (21) and (25) yields

$$\begin{aligned} & \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} (\alpha_i x_j - \alpha_j p_i) \\ + & \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} \alpha_j(p_i - x_i) \\ = & \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} (\alpha_i x_j - \alpha_j x_i). \end{aligned} \quad (26)$$

Observe that the sum of expressions (23) and (26) coincides with  $\gamma_i$ , since in (26) we sum over all neighbourswitches in which player  $i$  is involved as the back player that give rise to positive cost savings. We conclude that

$$\delta_i = \gamma_i - \sum_{j \in N: j > i \text{ and } \sigma^{-1}(j) < \sigma^{-1}(i)} \alpha_j(p_i - x_i) + \sum_{j \in N: j < i \text{ and } \sigma^{-1}(j) > \sigma^{-1}(i)} \alpha_i(p_j - x_j). \quad \square$$

The following lemma states that if a coalition  $S \subseteq N$  consists of several components, then the optimal schedule of  $S$  is also optimal for the last component. This result is logical since the jobs from other components do not benefit from the crashing of a job of the last component.

**Lemma A.1** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a sequencing situation with controllable processing times. Let  $S \subseteq N$  consist of  $t \geq 2$  components. Let  $S_t$  be the last component. Then every optimal schedule for  $S$  restricted to  $S_t$  is optimal for  $S_t$ .

The proof is omitted since it is trivial.

**Proof of Theorem 3.2:** Let  $(N, \sigma_0, \alpha, \beta, p, \bar{p})$  be a cps situation and let  $(N, v)$  be the corresponding game. Let  $1 \leq j \leq n$  and let  $\sigma : \{1, \dots, n\} \rightarrow N$  be such that  $\sigma(i) = n + 1 - i$  for all  $1 \leq i < j$ ,  $\sigma(i) = n + 2 - i$  for all  $j < i \leq n$  and  $\sigma(j) = 1$ . We need to show that for all  $0 \leq i \leq k$  and all  $S \subseteq N \setminus [\sigma(k), \sigma]$  expression (2) holds.

So let  $0 \leq i \leq k$  and let  $S \subseteq N \setminus [\sigma(k), \sigma]$ . We assume that  $1 \leq i < k$  and that  $S \neq \emptyset$ , since for  $i = 0$  expression (2) trivially holds because of superadditivity. Note that since  $S \neq \emptyset$ , it holds that  $k \neq n$ .

Because of the structure of  $\sigma$ , it holds that  $[\sigma(k), \sigma]$  and  $[\sigma(i), \sigma]$  both consist of at most two connected components. In particular, they consist of possibly job 1 and a tail of  $\sigma_0$ . If  $1 \in [\sigma(k), \sigma]$ , then we will denote the first component by  $K_1 = \{1\}$  and the second by  $K_2 = [\sigma(k), \sigma] \setminus \{1\}$ . If  $1 \notin [\sigma(k), \sigma]$ , then there is one component which we will denote by  $K_2$  and we define  $K_1 = \emptyset$ . Similarly, if  $1 \in [\sigma(i), \sigma]$ , then  $I_1 = \{1\}$  and  $I_2 = [\sigma(i), \sigma] \setminus \{1\}$ . If  $1 \notin [\sigma(i), \sigma]$ , then there is one component which we will denote by  $I_2$  and we define  $I_1 = \emptyset$ .

Denote the optimal processing schedule of  $[\sigma(k), \sigma]$  by  $(\tau^{[\sigma(k), \sigma]}, p^{[\sigma(k), \sigma]})$ , and the optimal processing schedule of  $[\sigma(i), \sigma] \cup S$  by  $(\tau^{[\sigma(i), \sigma] \cup S}, p^{[\sigma(i), \sigma] \cup S})$ . We will create suboptimal schedules for coalitions  $[\sigma(k), \sigma] \cup S$  and  $[\sigma(i), \sigma]$ , depending on the optimal schedules of coalitions  $[\sigma(i), \sigma] \cup S$  and  $[\sigma(k), \sigma]$ . In particular we "allocate" the processing times of coalitions  $[\sigma(i), \sigma] \cup S$  and  $[\sigma(k), \sigma]$  to coalitions  $[\sigma(i), \sigma]$  and  $[\sigma(k), \sigma] \cup S$ . In this way we obtain suboptimal processing schedules. The profit obtained at these suboptimal processing schedules give lowerbounds for  $v([\sigma(k), \sigma] \cup S)$  and  $v([\sigma(i), \sigma])$ . We will show that the sum of these lowerbounds exceeds the sum of  $v([\sigma(i), \sigma] \cup S)$  and  $v([\sigma(k), \sigma])$ . We distinguish between two cases.

**Case 1:**  $I_2 \subsetneq K_2$ .

Before we will obtain our suboptimal schedules, we first derive expressions for  $v([\sigma(k), \sigma])$  and  $v([\sigma(i), \sigma] \cup S)$ . It holds that

$$\begin{aligned} & v([\sigma(k), \sigma]) \\ &= \sum_{l \in K_1} ((\sum_{m \in [\sigma(k), \sigma]} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(k), \sigma]}) \end{aligned} \quad (27)$$

$$+ v(K_2). \quad (28)$$

The cost savings of  $[\sigma(k), \sigma]$  can be divided into two parts. The cost savings obtained by a possible crash of job 1 equal (27). Note that if  $1 \notin [\sigma(k), \sigma]$ , then  $K_1 = \emptyset$  and expression (27) is zero by definition. The other cost savings that can be obtained by interchanging and crashing jobs of  $K_2$  equal (28) according to Lemma A.1.

Since  $I_2$  and  $K_2$  are both tails of  $\sigma_0$ ,  $I_2 \subsetneq K_2$  and  $S \cap K_2 = \emptyset$ , we conclude that  $S$  is not connected to  $I_2$ . Denote the components of  $[\sigma(i), \sigma] \cup S$  by  $S_1, \dots, S_{t+1}$ , for  $t \geq 1$ . So  $S_{t+1} = I_2$  and if  $1 \in [\sigma(i), \sigma]$ , then  $1 \in S_1$ . Therefore

$$\begin{aligned} & v([\sigma(i), \sigma] \cup S) \\ &= \sum_{l \in I_1} ((\sum_{m \in [\sigma(i), \sigma] \cup S} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(i), \sigma] \cup S}) \end{aligned} \quad (29)$$

$$+ \sum_{l \in S} ((\sum_{m \in S \cup I_2: m \geq l} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(i), \sigma] \cup S}) \quad (30)$$

$$+ \sum_{l, m \in S_1: l < m} (\alpha_m p_l^{[\sigma(i), \sigma] \cup S} - \alpha_l p_m^{[\sigma(i), \sigma] \cup S})_+ \quad (31)$$

$$+ \sum_{a=2}^t (\sum_{l, m \in S_a: l < m} (\alpha_m p_l^{[\sigma(i), \sigma] \cup S} - \alpha_l p_m^{[\sigma(i), \sigma] \cup S})_+) \quad (32)$$

$$+ v(I_2). \quad (33)$$

Expression (29) denotes the cost savings obtained by a possible crash of job 1, if  $1 \in [\sigma(i), \sigma]$ . The cost savings obtained by crashing the jobs in  $S$  equal (30). Expressions (31) and (32) are the cost savings obtained by switching the jobs in  $S \cup I_1$ . Because of Lemma A.1 the cost savings obtained by crashing and rearranging jobs in  $I_2$  can be expressed as (33).

Now we will create suboptimal schedules  $(\pi^{[\sigma(i), \sigma]}, p^{[\sigma(i), \sigma]})$  and  $(\pi^{[\sigma(k), \sigma] \cup S}, p^{[\sigma(k), \sigma] \cup S})$  for coalitions  $[\sigma(i), \sigma]$  and  $[\sigma(k), \sigma] \cup S$  respectively. Let

$$p_l^{[\sigma(i), \sigma]} = \begin{cases} p_l^{[\sigma(i), \sigma] \cup S}, & \text{if } l \in I_2; \\ \max\{p_l^{[\sigma(i), \sigma] \cup S}, p_l^{[\sigma(k), \sigma]}\}, & \text{if } l = 1; \\ p_l, & \text{if } l \in N \setminus [\sigma(i), \sigma], l \neq 1. \end{cases}$$

Note that if  $1 \notin [\sigma(i), \sigma]$ , then it necessarily holds that  $1 \notin [\sigma(k), \sigma]$  or that  $1 \notin [\sigma(i), \sigma] \cup S$ , because  $[\sigma(k), \sigma] \cap S = \emptyset$ . This implies that if  $1 \notin [\sigma(i), \sigma]$ , then  $p_1^{[\sigma(k), \sigma]} = p_1$  or that  $p_1^{[\sigma(i), \sigma] \cup S} = p_1$ , and



therefore that  $p_1^{[\sigma(i),\sigma]} = p_1$ . We conclude that the processing times  $p^{[\sigma(i),\sigma]}$  satisfy the admissibility constraints (A1) and (A2).

Furthermore let  $\pi^{[\sigma(i),\sigma]}$  be obtained from  $\sigma_0$  by rearranging the jobs of  $[\sigma(i),\sigma]$  according to the Smith-rule using processing times  $p^{[\sigma(i),\sigma]}$ , taking of course into account the admissibility constraint (A3). Note that this schedule restricted to  $I_2$  is an optimal schedule for  $I_2$  according to Lemma A.1. This yields

$$v([\sigma(i),\sigma]) \geq \sum_{l \in I_1} ((\sum_{m \in [\sigma(i),\sigma]} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(i),\sigma]}) \quad (34)$$

$$+ v(I_2). \quad (35)$$

Similarly, let

$$p_l^{[\sigma(k),\sigma] \cup S} = \begin{cases} p_l^{[\sigma(k),\sigma]}, & \text{if } l \in K_2; \\ p_l^{[\sigma(i),\sigma] \cup S}, & \text{if } l \in S, l \neq 1; \\ \min\{p_l^{[\sigma(i),\sigma] \cup S}, p_l^{[\sigma(k),\sigma]}\}, & \text{if } l = 1; \\ p_l, & \text{if } l \in N \setminus ([\sigma(k),\sigma] \cup S), l \neq 1. \end{cases}$$

Observe that if  $1 \notin [\sigma(k),\sigma] \cup S$ , then  $1 \notin [\sigma(i),\sigma] \cup S$  and  $1 \notin [\sigma(k),\sigma]$ . Hence, if  $1 \notin [\sigma(k),\sigma] \cup S$ , then it holds that  $p_1^{[\sigma(k),\sigma]} = p_1$  and  $p_1^{[\sigma(i),\sigma] \cup S} = p_1$ , and therefore that  $p_1^{[\sigma(k),\sigma] \cup S} = p_1$ . We conclude that the processing times  $p^{[\sigma(k),\sigma] \cup S}$  satisfy the admissibility constraints (A1) and (A2).

Now using these processing times, let  $\pi^{[\sigma(k),\sigma] \cup S}$  be the order obtained from  $\sigma_0$  by interchanging the jobs according to the Smith-rule, while of course taking into account the admissibility constraint (A3). However, only interchange two jobs if both jobs are in  $K_2$ , or both jobs are in  $S \cup I_1$ . This last condition is a technical detail in order to keep the number of terms of our lowerbound for the cost savings of coalition  $[\sigma(k),\sigma] \cup S$  more manageable. Observe that this restriction only lowers our lowerbound of  $v([\sigma(k),\sigma] \cup S)$ . Again note, by Lemma A.1, that this processing schedule restricted to  $K_2$  is optimal for  $K_2$ . This yields

$$v([\sigma(k),\sigma] \cup S) \geq \sum_{l \in K_1} ((\sum_{m \in [\sigma(k),\sigma] \cup S} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(k),\sigma] \cup S}) \quad (36)$$

$$+ \sum_{l \in S} ((\sum_{m \in S \cup K_2: m \geq l} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(k),\sigma] \cup S}) \quad (37)$$

$$+ \sum_{l, m \in S_1: l < m} (\alpha_m p_l^{[\sigma(k),\sigma] \cup S} - \alpha_l p_m^{[\sigma(k),\sigma] \cup S})_+ \quad (38)$$

$$+ \sum_{a=2}^t (\sum_{l, m \in S_a: l < m} (\alpha_m p_l^{[\sigma(k),\sigma] \cup S} - \alpha_l p_m^{[\sigma(k),\sigma] \cup S})_+) \quad (39)$$

$$+ v(K_2). \quad (40)$$

Now first observe that expressions (35) and (33) coincide, as well as expressions (40) and (28). Furthermore expression (37) exceeds expression (30) since  $I_2 \subset K_2$  and  $p_l^{[\sigma(k),\sigma] \cup S} = p_l^{[\sigma(i),\sigma] \cup S}$  for all  $l \in S$ . It also holds that expression (32) coincides with expression (39) because  $p_l^{[\sigma(k),\sigma] \cup S} = p_l^{[\sigma(i),\sigma] \cup S}$  for all  $l \in S$ . We conclude that showing (2) boils down to showing that the sum of expressions (34), (36) and (38) exceeds the sum of expressions (27), (29) and (31). We now distinguish between three subcases.

**Subcase 1a:**  $1 \notin [\sigma(k),\sigma]$ .

This implies that  $K_1 = \emptyset$ , and thus that  $I_1 = \emptyset$ . Therefore it holds that expressions (27), (29), (34) and (36) all are equal to zero. Hence, it is sufficient to show that (38) exceeds (31) in order to show that (2) holds. Because  $I_1 = \emptyset$  it follows that  $1 \notin S_1$ . Therefore it holds that  $p_j^{[\sigma(i),\sigma] \cup S} = p_j^{[\sigma(k),\sigma] \cup S}$  for all  $j \in S_1$ . We conclude that (38) and (31) coincide.

**Subcase 1b:**  $1 \in [\sigma(k), \sigma]$  and  $p_1^{[\sigma(k),\sigma] \cup S} = p_1^{[\sigma(i),\sigma] \cup S}$ .

Because  $p_1^{[\sigma(k),\sigma] \cup S} = p_1^{[\sigma(i),\sigma] \cup S}$  it follows from the definition of  $p_1^{[\sigma(k),\sigma] \cup S}$  that  $p_1^{[\sigma(i),\sigma] \cup S} \leq p_1^{[\sigma(k),\sigma]}$ . We conclude that it must hold that  $p_1^{[\sigma(i),\sigma]} = p_1^{[\sigma(k),\sigma]}$ . Note that it now holds that (38) and (31) coincide, since  $p_j^{[\sigma(i),\sigma] \cup S} = p_j^{[\sigma(k),\sigma] \cup S}$  for all  $j \in S_1$ . So showing that (2) is satisfied, boils down to showing that the sum of expressions (34) and (36) exceeds the sum of expressions (27) and (29).

Now first suppose that  $p_1^{[\sigma(k),\sigma]} = p_1$ . Then it holds that expressions (27) is equal to zero. Since  $p_1^{[\sigma(i),\sigma]} = p_1^{[\sigma(k),\sigma]}$  it follows that expression (34) is equal to zero as well. Also we have that expression (36) exceeds expression (29), since  $p_1^{[\sigma(k),\sigma] \cup S} = p_1^{[\sigma(i),\sigma] \cup S}$  and  $[\sigma(i),\sigma] \subset [\sigma(k),\sigma]$ . So expression (2) is satisfied in this case.

Secondly, suppose that  $p_1^{[\sigma(k),\sigma]} < p_1$ . Since optimal processing times can only take two values it holds that  $p_1^{[\sigma(k),\sigma]} = \bar{p}_1$ . Since by assumption of subcase 1b it holds that  $p_1^{[\sigma(k),\sigma] \cup S} = \min\{p_1^{[\sigma(k),\sigma]}, p_1^{[\sigma(i),\sigma] \cup S}\} = p_1^{[\sigma(i),\sigma] \cup S}$  it follows that  $p_1^{[\sigma(i),\sigma] \cup S} = \bar{p}_1$ . We conclude that  $p_1^{[\sigma(i),\sigma]} = \max\{p_1^{[\sigma(k),\sigma]}, p_1^{[\sigma(i),\sigma] \cup S}\} = \max\{\bar{p}_1, \bar{p}_1\} = \bar{p}_1$ .

Observe that because  $1 \in [\sigma(k), \sigma]$ , that  $1 \notin S$ . Since  $p_1^{[\sigma(i),\sigma] \cup S} < p_1$ , this implies that  $1 \in [\sigma(i), \sigma]$ . Thus,  $K_1 = I_1 = \{1\}$ . Hence it follows for the sum of expressions (27) and (29) that

$$\begin{aligned} & \sum_{l \in K_1} ((\sum_{m \in [\sigma(k), \sigma]} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(k), \sigma]}) + \sum_{l \in I_1} ((\sum_{m \in [\sigma(i), \sigma] \cup S} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(i), \sigma] \cup S}) \\ &= ((\sum_{m \in [\sigma(k), \sigma]} \alpha_m) - \beta_1)(p_1 - \bar{p}_1) + ((\sum_{m \in [\sigma(i), \sigma] \cup S} \alpha_m) - \beta_1)(p_1 - \bar{p}_1) \\ &= ((\sum_{m \in [\sigma(k), \sigma] \cup S} \alpha_m) - \beta_1)(p_1 - \bar{p}_1) + ((\sum_{m \in [\sigma(i), \sigma]} \alpha_m) - \beta_1)(p_1 - \bar{p}_1), \end{aligned}$$

where the first equality is satisfied since  $K_1 = I_1 = \{1\}$  and  $p_1^{[\sigma(k),\sigma]} = p_1^{[\sigma(i),\sigma] \cup S} = \bar{p}_1$ . Note that this last expression is equal to the sum of expressions (34) and (36) since  $p_1^{[\sigma(i),\sigma]} = p_1^{[\sigma(k),\sigma] \cup S} = \bar{p}_1$  and  $K_1 = I_1 = \{1\}$ . We conclude that (2) is satisfied.

**Subcase 1c:**  $1 \in [\sigma(k), \sigma]$  and  $p_1^{[\sigma(k),\sigma] \cup S} < p_1^{[\sigma(i),\sigma] \cup S}$ .

We now necessarily have that  $p_1^{[\sigma(k),\sigma] \cup S} = p_1^{[\sigma(k),\sigma]}$  and that  $p_1^{[\sigma(i),\sigma]} = p_1^{[\sigma(i),\sigma] \cup S}$ . Since optimal processing times can only take two values it follows that  $p_1^{[\sigma(k),\sigma] \cup S} = p_1^{[\sigma(k),\sigma]} = \bar{p}_1$  and that  $p_1^{[\sigma(i),\sigma]} = p_1^{[\sigma(i),\sigma] \cup S} = p_1$ . Therefore expressions (29) and (34) both are equal to zero. So showing that (2) holds boils down to showing that the sum of expressions (36) and (38) exceeds the sum of expressions (27) and (31). First observe for expression (36) that

$$\begin{aligned} & \sum_{l \in K_1} ((\sum_{m \in [\sigma(k), \sigma] \cup S} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(k), \sigma] \cup S}) \\ &= ((\sum_{m \in [\sigma(k), \sigma] \cup S} \alpha_m) - \beta_1)(p_1 - \bar{p}_1) \end{aligned}$$

$$\geq ((\sum_{m \in [\sigma(k), \sigma]} \alpha_m) - \beta_1)(p_1 - \bar{p}_1) \quad (41)$$

$$+ ((\sum_{m \in S_1 \setminus \{1\}} \alpha_m)(p_1 - \bar{p}_1), \quad (42)$$

where the equality holds because  $K_1 = \{1\}$  and because  $p_1^{[\sigma(k), \sigma] \cup S} = \bar{p}_1$ . The inequality holds since  $(S_1 \setminus \{1\}) \subseteq S$ . Also observe for expression (38) that

$$\begin{aligned} & \sum_{l, m \in S_1: l < m} (\alpha_m p_l^{[\sigma(k), \sigma] \cup S} - \alpha_l p_m^{[\sigma(k), \sigma] \cup S})_+ \\ &= \sum_{l, m \in S_1 \setminus \{1\}: l < m} (\alpha_m p_l^{[\sigma(k), \sigma] \cup S} - \alpha_l p_m^{[\sigma(k), \sigma] \cup S})_+ \end{aligned} \quad (43)$$

$$+ \sum_{m \in S_1 \setminus \{1\}} (\alpha_m \bar{p}_1 - \alpha_1 p_m^{[\sigma(k), \sigma] \cup S})_+. \quad (44)$$

For the equality we have used that  $p_1^{[\sigma(k), \sigma] \cup S} = \bar{p}_1$ . Since  $K_1 = \{1\}$  and  $p_1^{[\sigma(k), \sigma]} = \bar{p}_1$ , it follows that expression (41) coincides with expression (27). Therefore it is now sufficient to show that the sum of expressions (42), (43) and (44) exceeds expression (31). If it holds that  $1 \notin S_1$ , then it holds that expression (43) coincides with expression (31) since it holds that  $p_m^{[\sigma(k), \sigma] \cup S} = p_m^{[\sigma(i), \sigma] \cup S}$  for all  $m \in S_1 \setminus \{1\}$ . Therefore assume that  $1 \in S_1$ . For the sum of expressions (42) and (44) we have that

$$\begin{aligned} & ((\sum_{m \in S_1 \setminus \{1\}} \alpha_m)(p_1 - \bar{p}_1) + \sum_{m \in S_1 \setminus \{1\}} (\alpha_m \bar{p}_1 - \alpha_1 p_m^{[\sigma(k), \sigma] \cup S})_+ \\ & \geq \sum_{m \in S_1 \setminus \{1\}} (\alpha_m p_1 - \alpha_1 p_m^{[\sigma(k), \sigma] \cup S})_+ \\ &= \sum_{m \in S_1 \setminus \{1\}} (\alpha_m p_1^{[\sigma(i), \sigma] \cup S} - \alpha_1 p_m^{[\sigma(i), \sigma] \cup S})_+, \end{aligned} \quad (45)$$

The inequality holds because of Lemma 3.2 by taking  $a_1 = \alpha_1$ ,  $a_2 = \alpha_m$ ,  $q_1 = p_1$ ,  $\bar{q}_1 = \bar{p}_1$  and  $q_2 = p_m^{[\sigma(k), \sigma] \cup S}$ . The equality is satisfied because  $p_1^{[\sigma(i), \sigma] \cup S} = p_1$  and  $p_m^{[\sigma(k), \sigma] \cup S} = p_m^{[\sigma(i), \sigma] \cup S}$  for all  $m \in S_1 \setminus \{1\}$ . Now observe that the sum of expressions (43) and (45) coincides with expression (31) since  $p_m^{[\sigma(k), \sigma] \cup S} = p_m^{[\sigma(i), \sigma] \cup S}$  for all  $m \in S_1 \setminus \{1\}$  and our assumption that  $1 \in S_1$ . We conclude that (2) is satisfied.

**Case 2:**  $I_2 = K_2$ .

Since we have assumed that  $i < k$  and hence that  $[\sigma(i), \sigma] \neq [\sigma(k), \sigma]$  we conclude, using the structure of  $\sigma$ , that  $I_1 = \emptyset$  and  $K_1 = \{1\}$ . Denote the components of  $[\sigma(i), \sigma] \cup S$  by  $S_1, \dots, S_t$ , with  $t \geq 1$ . Note that it might hold that  $I_2 \subsetneq S_t$ . We have that

$$\begin{aligned} & v([\sigma(k), \sigma]) \\ &= ((\sum_{m \in [\sigma(k), \sigma]} \alpha_m) - \beta_1)(p_1 - p_1^{[\sigma(k), \sigma]}) \end{aligned} \quad (46)$$

$$+ v(K_2) \quad (47)$$

and that

$$\begin{aligned} & v([\sigma(i), \sigma] \cup S) \\ &= \sum_{l \in [\sigma(i), \sigma] \cup S} ((\sum_{m \in [\sigma(i), \sigma] \cup S: m \geq l} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(i), \sigma] \cup S}) \end{aligned} \quad (48)$$

$$+ \sum_{a=1}^t (\sum_{l, m \in S_a: l < m} (\alpha_m p_l^{[\sigma(i), \sigma] \cup S} - \alpha_l p_m^{[\sigma(i), \sigma] \cup S})_+). \quad (49)$$

First we note, because  $I_1 = \emptyset$ , that  $v([\sigma(i), \sigma]) = v(I_2) = v(K_2)$ . So showing that (2) is satisfied boils down to showing that the sum of expressions (46), (48) and (49) is exceeded by  $v([\sigma(k), \sigma] \cup S)$ . We will obtain a lowerbound of  $v([\sigma(k), \sigma] \cup S)$  by creating a suboptimal schedule for coalition  $[\sigma(k), \sigma] \cup S$ , which we will denote by  $(\pi^{[\sigma(k), \sigma] \cup S}, p^{[\sigma(k), \sigma] \cup S})$ . Let the processing times be given by

$$p_l^{[\sigma(k), \sigma] \cup S} = \begin{cases} p_l^{[\sigma(i), \sigma] \cup S}, & \text{if } l \in [\sigma(i), \sigma] \cup S; \\ p_l^{[\sigma(k), \sigma]}, & \text{if } l = 1; \\ p_l, & \text{if } l \in N \setminus ([\sigma(k), \sigma] \cup S). \end{cases}$$

Let  $\pi^{[\sigma(k), \sigma] \cup S}$  be constructed by rearranging the jobs of coalition  $[\sigma(k), \sigma] \cup S$  using the Smith-rule and our suboptimal processing times. However, only switch jobs if both are in  $[\sigma(i), \sigma] \cup S$ . We can conclude for the cost savings of  $[\sigma(k), \sigma] \cup S$  that

$$\begin{aligned} & v([\sigma(k), \sigma] \cup S) \\ & \geq ((\sum_{m \in [\sigma(k), \sigma] \cup S} \alpha_m) - \beta_1)(p_1 - p_1^{[\sigma(k), \sigma] \cup S}) \end{aligned} \quad (50)$$

$$+ \sum_{l \in ([\sigma(k), \sigma] \cup S) \setminus \{1\}} ((\sum_{m \in [\sigma(k), \sigma] \cup S: m \geq l} \alpha_m) - \beta_l)(p_l - p_l^{[\sigma(k), \sigma] \cup S}) \quad (51)$$

$$+ \sum_{a=1}^t (\sum_{l, m \in S_a: l < m} (\alpha_m p_l^{[\sigma(k), \sigma] \cup S} - \alpha_l p_m^{[\sigma(k), \sigma] \cup S})_+). \quad (52)$$

Expression (46) is exceeded by expression (50) since  $p_1^{[\sigma(k), \sigma] \cup S} = p_1^{[\sigma(i), \sigma]}$ . Note that expression (51) coincides with expression (48) since  $([\sigma(k), \sigma] \cup S) \setminus \{1\} = [\sigma(i), \sigma] \cup S$  and because  $p_l^{[\sigma(k), \sigma] \cup S} = p_l^{[\sigma(i), \sigma] \cup S}$  for all  $l \in [\sigma(i), \sigma] \cup S$ . Furthermore we have that expressions (49) and (52) coincide because  $p_l^{[\sigma(i), \sigma] \cup S} = p_l^{[\sigma(k), \sigma] \cup S}$  for all  $l \in [\sigma(i), \sigma] \cup S$ . We conclude that the sum of  $v([\sigma(k), \sigma] \cup S)$  and  $v([\sigma(i), \sigma])$  exceeds the sum of  $v([\sigma(i), \sigma] \cup S)$  and  $v([\sigma(k), \sigma])$ . This ends the proof.  $\square$

**Proof of Lemma 3.3:** Let  $S, T \subseteq N$  with  $S \cap T = \{i\}$  for some  $i \in N$ . We first introduce some notation. Denote the connected components of  $S$  by  $S_k$ ,  $k = 1, \dots, a$  and those of  $T$  by  $T_l$ ,  $l = 1, \dots, b$ . Let  $S_c$ ,  $1 \leq c \leq a$  be such that  $i \in S_c$ , and let  $T_d$ ,  $1 \leq d \leq b$  be such that  $i \in T_d$ . Finally let  $S_c^1 = \{m \in S_c : m < i\}$ ,  $S_c^2 = \{m \in S_c : m > i\}$ ,  $T_d^1 = \{m \in T_d : m < i\}$  and  $T_d^2 = \{m \in T_d : m > i\}$ . Note that since  $S \cap T = \{i\}$  it holds that  $S_c^1 \cap T_d^1 = \emptyset$  and that  $S_c^2 \cap T_d^2 = \emptyset$ . Let the connected components of  $S \cup T$  be denoted by  $B_k$ ,  $k = 1, \dots, e$ . Now let  $(\sigma^S, p^S) \in A^S$  be an optimal processing schedule for coalition  $S$  and  $(\sigma^T, p^T) \in A^T$  be an optimal processing schedule of coalition  $T$ . It holds that

$$v(S) = \sum_{j \in S} ((\sum_{k \in S: k \geq j} \alpha_k) - \beta_j)(p_j - p_j^S) \quad (53)$$

$$+ \sum_{k=1}^a (\sum_{l, m \in S_k: l < m} (\alpha_m p_l^S - \alpha_l p_m^S)_+). \quad (54)$$

The cost savings for coalition  $S$  can be split into two components. The first component, (53), are the (possibly negative) cost savings that are obtained by crashing some of the jobs. Expression (54) contains the cost savings obtained by interchanging the jobs. Similarly it holds that

$$v(T) = \sum_{j \in T} ((\sum_{k \in T: k \geq j} \alpha_k) - \beta_j)(p_j - p_j^T) \quad (55)$$

$$+ \sum_{k=1}^b (\sum_{l, m \in T_k: l < m} (\alpha_m p_l^T - \alpha_l p_m^T)_+). \quad (56)$$

We will now construct a suboptimal schedule for coalition  $S \cup T$ . We will show that the cost savings obtained at this suboptimal schedule exceed the sum of  $v(S)$  and  $v(T)$ .

Consider the suboptimal processing schedule  $(\sigma^{S \cup T}, p^{S \cup T}) \in A^{S \cup T}$  for coalition  $S \cup T$ , where

$$p_j^{S \cup T} = \begin{cases} p_j^S, & \text{if } j \in S \setminus \{i\}; \\ p_j^T, & \text{if } j \in T \setminus \{i\}; \\ \min\{p_j^S, p_j^T\}, & \text{if } j = i; \\ p_j, & \text{if } j \in N \setminus (S \cup T). \end{cases}$$

Furthermore,  $\sigma^{S \cup T}$  is the order obtained by rearranging the jobs of  $S \cup T$  according to the Smith-rule using as processing times  $p^{S \cup T}$  and taking into account the admissibility constraint (A3). This yields

$$\begin{aligned} v(S \cup T) &\geq \sum_{j \in S \cup T} ((\sum_{k \in S \cup T: k \geq j} \alpha_k) - \beta_j)(p_j - p_j^{S \cup T}) \\ &+ \sum_{k=1}^e (\sum_{l, m \in B_k: l < m} (\alpha_m p_l^{S \cup T} - \alpha_l p_m^{S \cup T})_+) \\ &= \sum_{j \in S \setminus \{i\}} ((\sum_{k \in S \cup T: k \geq j} \alpha_k) - \beta_j)(p_j - p_j^S) \end{aligned} \quad (57)$$

$$+ \sum_{j \in T \setminus \{i\}} ((\sum_{k \in S \cup T: k \geq j} \alpha_k) - \beta_j)(p_j - p_j^T) \quad (58)$$

$$+ ((\sum_{k \in S \cup T: k \geq i} \alpha_k) - \beta_i)(p_i - p_i^{S \cup T}) \quad (59)$$

$$+ \sum_{k=1}^e (\sum_{l, m \in B_k: l < m} (\alpha_m p_l^{S \cup T} - \alpha_l p_m^{S \cup T})_+). \quad (60)$$

We will now distinguish between three cases in order to prove our inequality.

**Case 1:**  $p_i^S = p_i^T = p_i$ .

First note that it holds that  $p_j^{S \cup T} = p_j^S$  for all  $j \in S$  and that  $p_j^{S \cup T} = p_j^T$  for all  $j \in T$ . It holds that expression (59) is equal to zero because  $p_i^{S \cup T} = p_i$ . Furthermore, expression (57) exceeds expression (53), as well as expression (58) exceeds expression (55), since  $p_i^S = p_i^T = p_i$ . So showing that  $v(S) + v(T) \leq v(S \cup T)$  boils down to showing that expression (60) exceeds the sum of expressions (54) and (56). Let  $j, h \in S_k$  for some  $1 \leq k \leq a$ , i.e. let  $j, h$  appear in a connected component of  $S$ . It follows that they also appear in a connected component of  $S \cup T$ , i.e. there is a  $1 \leq g \leq e$  with  $j, h \in B_g$ . We conclude that the term in (54) dealing with  $j$  and  $h$  also appears in (60). Similarly, each pair  $j, h \in T_l$ , for some  $1 \leq l \leq b$ , appears in a connected component of  $S \cup T$ . Therefore we also have that the term in (56) dealing with  $j, h$  also appears in (60). Observe that each pair in  $S$  is not in  $T$ , and that each pair in  $T$  is not in  $S$ , because  $|S \cap T| = 1$ . Thus it holds, since  $p_j^{S \cup T} = p_j^S$  for all  $j \in S$ ,  $p_j^{S \cup T} = p_j^T$  for all  $j \in T$  and the nonnegativity of every term in (60),

that (60) exceeds the sum of (54) and (56).

**Case 2:**  $p_i^S = p_i^T = \bar{p}_i$ .

Note that it holds by definition of  $p_i^{S \cup T}$  that  $p_i^{S \cup T} = p_i^S = p_i^T$ . Observe that this implies that  $p_j^{S \cup T} = p_j^S$  for all  $j \in S$  and  $p_j^{S \cup T} = p_j^T$  for all  $j \in T$ . First we will develop a lowerbound for (59):

$$\begin{aligned}
& ((\sum_{k \in S \cup T: k \geq i} \alpha_k) - \beta_i)(p_i - p_i^{S \cup T}) \\
&= ((\sum_{k \in S: k \geq i} \alpha_k) - \beta_i)(p_i - p_i^{S \cup T}) \\
&+ ((\sum_{k \in T: k \geq i} \alpha_k) - \beta_i)(p_i - p_i^{S \cup T}) \\
&+ (\beta_i - \alpha_i)(p_i - p_i^{S \cup T}) \\
&\geq ((\sum_{k \in S: k \geq i} \alpha_k) - \beta_i)(p_i - p_i^S) \tag{61}
\end{aligned}$$

$$+ ((\sum_{k \in T: k \geq i} \alpha_k) - \beta_i)(p_i - p_i^T) \tag{62}$$

where the inequality is satisfied since  $\beta_i \geq \alpha_i$ , and by assumption  $p_i^{S \cup T} = p_i^S = p_i^T$ . Observe that the sum of expressions (57) and (61) exceeds expression (53) and that the sum of expressions (58) and (62) exceeds expression (55). Hence, showing that  $v(S) + v(T) \leq v(S \cup T)$  boils down to showing that expression (60) exceeds the sum of expressions (54) and (56). For this last statement we refer to Case 1, where we already showed this inequality. Note that we can refer to Case 1, since in both cases we have that  $p_j^{S \cup T} = p_j^S$  for all  $j \in S$  and  $p_j^{S \cup T} = p_j^T$  for all  $j \in T$ .

**Case 3:**  $p_i^S \neq p_i^T$ .

Without loss of generality assume that  $p_i^S < p_i^T$ , or equivalently that  $p_i^S = \bar{p}_i$  and  $p_i^T = p_i$ . Hence we have, by definition of  $p_i^{S \cup T}$ , that  $p_i^{S \cup T} = p_i^S = \bar{p}_i$ . Observe that we can obtain the following lowerbound for (59):

$$\begin{aligned}
& ((\sum_{k \in S \cup T: k \geq i} \alpha_k) - \beta_i)(p_i - p_i^{S \cup T}) \\
&= ((\sum_{k \in S: k \geq i} \alpha_k) - \beta_i)(p_i - p_i^S) \\
&+ (\sum_{k \in T: k > i} \alpha_k)(p_i - p_i^S) \\
&\geq ((\sum_{k \in S: k \geq i} \alpha_k) - \beta_i)(p_i - p_i^S) \tag{63}
\end{aligned}$$

$$+ (\sum_{m \in T_d^2} \alpha_m)(p_i - p_i^S), \tag{64}$$

where the equality holds because  $p_i^{S \cup T} = p_i^S$  and the inequality since  $T_d^2 \subset T$ . Observe that it holds that expression (58) exceeds expression (55), since  $p_i^T = p_i$ . Furthermore we have that the sum of expressions (57) and (63) exceeds expression (53). Therefore, showing that  $v(S) + v(T) \leq v(S \cup T)$

boils down to showing that the sum of expressions (60) and (64) exceeds the sum of expressions (54) and (56). Note that because  $|S \cap T| = 1$  we have the following lowerbound for (60):

$$\begin{aligned} & \sum_{k=1}^e \left( \sum_{l,m \in B_k: l < m} (\alpha_m p_l^{S \cup T} - \alpha_l p_m^{S \cup T})_+ \right) \\ & \geq \sum_{k=1}^a \left( \sum_{l,m \in S_k: l < m} (\alpha_m p_l^{S \cup T} - \alpha_l p_m^{S \cup T})_+ \right) \end{aligned} \quad (65)$$

$$+ \sum_{k=1}^b \left( \sum_{l,m \in T_k: l < m} (\alpha_m p_l^{S \cup T} - \alpha_l p_m^{S \cup T})_+ \right). \quad (66)$$

Now it holds that (65) coincides with (54), since  $p_l^{S \cup T} = p_l^S$  for all  $l \in S$ . So we only need to show that the sum of expressions (66) and (64) exceeds expression (56). Observe that for (66) we have that

$$\begin{aligned} & \sum_{k=1}^b \left( \sum_{l,m \in T_k: l < m} (\alpha_m p_l^{S \cup T} - \alpha_l p_m^{S \cup T})_+ \right) \\ & = \sum_{k=1}^b \left( \sum_{l,m \in T_k \setminus \{i\}: l < m} (\alpha_m p_l^{S \cup T} - \alpha_l p_m^{S \cup T})_+ \right) + \sum_{l \in T_d^1} (\alpha_i p_l^{S \cup T} - \alpha_l p_i^{S \cup T})_+ \\ & + \sum_{m \in T_d^2} (\alpha_m p_i^{S \cup T} - \alpha_i p_m^{S \cup T})_+ \\ & \geq \sum_{k=1}^b \left( \sum_{l,m \in T_k \setminus \{i\}: l < m} (\alpha_m p_l^T - \alpha_l p_m^T)_+ \right) + \sum_{l \in T_d^1} (\alpha_i p_l^T - \alpha_l p_i^T)_+ \end{aligned} \quad (67)$$

$$+ \sum_{m \in T_d^2} (\alpha_m p_i^S - \alpha_i p_m^T)_+, \quad (68)$$

where the inequality holds since  $p_j^{S \cup T} = p_j^T$  for all  $j \in T \setminus \{i\}$ ,  $p_i^{S \cup T} \leq p_i^T$  and  $p_i^{S \cup T} = p_i^S$ . Now adding (64) and (68) yields

$$\begin{aligned} & \left( \sum_{m \in T_d^2} \alpha_m \right) (p_i - p_i^S) + \sum_{m \in T_d^2} (\alpha_m p_i^S - \alpha_i p_m^T)_+ \\ & \geq \sum_{m \in T_d^2} (\alpha_m p_i - \alpha_i p_m^T)_+ \\ & = \sum_{m \in T_d^2} (\alpha_m p_i^T - \alpha_i p_m^T)_+, \end{aligned} \quad (69)$$

where the inequality holds because of Lemma 3.2 by taking  $a_1 = \alpha_i$ ,  $a_2 = \alpha_m$ ,  $q_1 = p_i$ ,  $\bar{q}_1 = p_i^S$  and  $q_2 = p_m^T$ . The equality is satisfied because  $p_i^T = p_i$ . Since the sum of expressions (67) and (69) coincides with expression (56) we conclude that  $v(S) + v(T) \leq v(S \cup T)$ .  $\square$